关联规则挖掘技术研究的新进展*)

刘君强[1,2] 孙晓莹[1] 潘云鹤[2]

(杭州商学院计算机系 杭州310035)[1] (浙江大学计算机科学与技术学院 杭州310027)[2]

摘 要 在数据库中发现频繁模式和关联规则是数据挖掘领域的最基本、最重要的问题。自从 Agrawal 的开创性工作以来,有关研究从未停止过。然而由于其内在的计算复杂性,这一问题并未完全解决。本文对关联规则的基本概念和类型、重要挖掘算法、特别是近年来相关研究的新进展做了全面综述与深入分析,并指出了未来的研究方向。

关键词 知识发现,数据挖掘,频繁模式,关联规则,综述

Survey on Association Rules Mining Technology

LIU Jun-Qiang[1,2] SUN Xiao-Ying[1] PAN Yun-He[2]

(Department of Computer Science, Hangzhou Institute of Commerce, Hangzhou 310035)[1]

(College of Computer Science and Technology, Zhejiang University, Hangzhou 310027)[2]

Abstract This paper discusses the discovery of frequent pattern and association rules in databases, which is the most fundamental and important problem in data mining area. Since the pioneering work of Agrawal, there has been an increase of research interest in this problem. Yet, it is not fully addressed because of the inherent computational complexity. We present an overview of the different kinds of algorithms and new developments of research in this area, and discuss the future direction of the research.

Keywords Knowledge discovery, Data mining, Frequent patterns, Association rules, Survery

1 引言

关联规则问题首先由 Agrawal 等[4]提出,是数据挖掘领域一个最基本、最重要的问题。关联规则定义在由一组称为事务的布尔型记录组成的数据库之上。每个记录包含了项目出现与否的信息。例如,在超市的交易数据库中,一个事务是一个顾客所买商品的集合。发现不同商品购买行为之间的关系很有用。这种关系可以表达为关联规则。关联规则可应用于消费者行为分析、商品货架设计、库存控制、商品促销、电子商务等。

挖掘关联规则的关键步骤是发现频繁模式集,简称频集。频集挖掘的最杰出算法是 Agrawal 等提出的 Apriori 算法。当频繁模式长度较短和数据集较稀疏、规模较小时,Apriori 的性能不错。但是,对于存在长模式、密集型、或海量数据集,Apriori 的时间效率和空间可伸缩性都面临挑战。因此,Apriori 以后有关频集和关联规则挖掘的研究一直很活跃,大致涉及三个方面:一是经典频集挖掘的高性能算法的研究,包括对 Apriori 的改进,以及探索新的挖掘方法;二是拓展频集的概念,提出相应的挖掘算法;三是拓展关联规则概念及应用范围,包括规则的价值评估、新的关联规则类型等。

2 频集与关联规则的基本概念

2.1 基本定义

经典的关联规则挖掘问题可以划分为两个子问题。第一个子问题是发现所有频繁模式组成的完全集,简称完全频集。

给定项目集 $I=\{i_1,i_2,\cdots,i_m\}$,文字 i_k 是一个项目。数据库 $T=\{(tid_1,t_1),(tid_2,t_2),\cdots,(tid_n,t_n)\}$ 是事务的集合,其中 tid_k 是事务的标识符,t_k 是组成事务的项目的集合,即 $t_k\subseteq I$。

模式 $p\subseteq I$ 被事务 (tid,t) 所包含,如果 $p\subseteq t$。T 对 p 的绝对支持率是 T 中包含 p 的事务数。支持率不小于用户给定最小支持率阈值的模式为频繁模式。

第二个子问题是抽取关联规则。如果 $x\cup y$ 是频繁模式,$x\rightarrow y$ 的可信度是 x 支持率与 $x\cup y$ 支持率的比值。可信度衡量 x 与 y 之间的关联性,当可信度不小于用户给定的最小可信度阈值时,$x\rightarrow y$ 称为关联规则。对于完全频集中的每个频繁模式 y 的任意子集 $x\subset y$,生成关联规则 $x\rightarrow(y-x)$,如果其可信度不小于阈值。

2.2 频集与关联规则的分类

频集与关联规则可以根据多种标准分类。根据数据类型,可分为布尔型与数量型;根据规则涉及的数据维数,分为单维与多维;根据规则集涉及的抽象层次,分为单层与多层;根据模式与规则间的相互关系,分为完全型、闭合型、最大型。频集与关联规则挖掘还可以根据应用范围,拓展为顺序模式、周期性片断、空间关联规则等问题。

3 完全频集与关联规则挖掘

3.1 Apriori 算法

Agrawal 提出的 Apriori 算法[5,6,23]是挖掘完全频集与关联规则,也就是单维单层布尔型关联规则的最具影响的算法。

Apriori 算法的基础是频繁模式的(反)单调性原则,即频繁模式的子模式必定是频繁的,而包含非频繁模式的超级模式必定是非频繁的。令 L_k 是长度为 k 的频繁模式集合,C_k 是长度为 k 的候选频繁模式的集合。C_1 就是 I,扫描 T 一遍,可求得 L_1。求长度为 $k+1(k\geq 1)$ 的频繁模式集 L_{k+1} 如下:

```
for(k=1;Lk!=∅;k++)do begin
  Ck+1=apriori-gen(Lk);
  for each transaction (tid,t)∈T do begin
    Ct=subset(Ck+1,t);
    for each candidate c∈Ct do c.support++;
```

*)基金项目:浙江省自然科学基金(GJ0302),浙江省教育厅科技计划(20020635)。

end
$L_{k+1}=\{c\in C_{k+1}\mid c.\text{support}\geq minsup\}$
end

首先，apriori-gen(L_k)由 join 和 prune 组成。join 步对每2个有 k－1个共同项目的长度 k 频繁模式进行连接，得到候选集 C'_{k+1}；prune 步根据(反)单调性原则，剪裁 C'_{k+1}，得 C_{k+1}。然后，扫描 T 一遍，确定每个事务(tid,t)中所含 k＋1候选频繁模式 subset(C_{k+1},t)，统计其中各个模式的支持率。再者，从 C_{k+1}中剔除支持率低于阈值者，得长度 k＋1的频繁模式集 L_{k+1}。

Apriori 的特点是：一、扫描数据库的次数等于最大频繁模式的长度；二、需要生成频繁模式的候选集。当支持率阈值较小而出现大量“长”关联规则时，Apriori 代价很高，容易出现组合爆炸。

3.2 对 Apriori 算法的改进

对于 Apriori 的改进主要在于控制候选集的规模或减少数据库扫描次数等几个方面。

(1)Park 等提出杂凑表技术[27]。根据 C_k 确定 C'_{k+1}，并用规模适当的 hash 存放 C'_{k+1}。在第 k 遍扫描数据库时，同时统计 C_k 和 hash 表中的 C'_{k+1}项目。在求出 L_k 的同时，hash 表中 C'_{k+1}的计数可用于进一步剪裁 C_{k+1}。

(2)Agrawal、Han、Park 等均提出减少扫描数据库事务记录的方法[7,16,27]。如果一个事务记录中不包含长度 k 的频繁模式，则不可能包含长度 $k+1$的频繁模式，因此可在以后的扫描中剔除。

(3)Savasere 等提出 Partition 算法[31]将数据库分割为若干个可调入内存的子库，分别求出各个子库的局部频繁模式，所有局部频繁模式的并集为全局频繁模式的候选集，最后一遍扫描数据库可最终求出全局频繁模式集。Partition 考虑的候选集比 Apriori 还要多，有可能加剧组合爆炸的问题。

(4)Toivonen 提出抽样法[32]。从数据库中随机抽取一个可调入内存的子集，采用一个略低的支持率阈值，求出该子集中的局部频繁模式，第二遍扫描数据库，求出局部频繁模式的全局支持率。该文还提出了确保全局频繁模式不被遗漏的机制。

(5)Brin 等提出动态模式计数法 DIC[10]。在同一遍数据库扫描过程中分段增加候选频繁模式集。DIC 在确定一个模式的所有子集都是频繁后，不久就开始其支持率的统计，而不是等到下一轮数据库扫描。

3.3 频集与关联规则挖掘研究的新进展

(1)Agarwal 等提出 TreeProjection 算法[1]，将频繁模式挖掘转化为逐步构造一种模式字典树的过程，与之相伴随的是将一个大型数据库投影为一系列子库的过程。它直接对数据库进行投影，投影子库可以用某种形式存放或直接扫描原始数据库临时产生，投影的策略有宽度优先、深度优先、混合投影等三种策略。

(2)Han 等提出了不生成候选集直接生成频繁模式的算法 FPGrowth[17]。其基本思想是将整个数据库压缩表达为 FP-Tree，将频繁模式挖掘过程转化为递归地产生“条件”子库及对应的“条件”FP-Tree 的过程。

(3)Liu 等提出伺机挖掘思想[20]，首次发现密集型数据子集的虚拟投影方法，以及稀疏型数据子集的非过滤投影方法，巧妙地解决了提高时间效率与节省存贮开销的矛盾；提出在挖掘过程中不断根据局部数据子集特性自动调整解空间搜索策略、决定数据子集表示形式、选择投影方法的启发式原则；设计实现的算法 OpportuneProject 性能远优于 Apriori、FPGrowth 等，特别是挖掘海量数据的性能要高若干数量级。

4 闭合频集与无冗余关联规则挖掘

4.1 频繁闭合模式与 A-Close 算法

挖掘完全频集往往很困难，因为可能的频繁模式数与数据库中项目数成指数关系。为此，Pasquier 等提出频繁闭合模式，它是不被其它具有相同支持率的模式所包含的频繁模式，其集合简称为闭合频集，从中可抽取出无冗余关联规则集。

Pasquier 等先后提出 Close 和 A-Close 算法[25,26]。A-Close 算法描述如下，核心是构造生成子集合。模式 p 是闭合模式 c 的生成子，如果 p 的闭包等于 c，且 p 的任意子集的闭包不等于 c。

$G_1.generator=L_1$;
$for(i=1;G_i.generator\neq\varnothing;i++)do$
  $G_{i+1}=AC\text{-}Generator(G_i)$;
$G=\cup G_i$;
$FC=AC\text{-}Closure(G)$;

长度为1的生成子集合 G_1就是频繁项目的集合 L_1。长度为 $i+1$的生成子集合 G_{i+1}用 G_i 来构造：首先执行 $join$ 操作，然后做第一次裁剪，即$\forall\ c\in G_{i+1},\exists\ s\subset c(i\text{-}subset),s\notin G_i\Rightarrow prune\ c\ from\ G_{i+1}$，这包括两种情况，一是 s 非频繁，则 c 也非频繁；二是 s 为某个生成子的闭包所包含，则 c 必为某个生成子的闭包所包含。接着扫描数据库，确定 G_{i+1}中各生成子的支持率。第二次裁剪，$\forall\ c\in G_{i+1},support(c)<minsup\Rightarrow prune\ c\ from\ G_{i+1}$，去掉非频繁生成子。最后，删除与其子集具有相同闭包的生成子，$\forall\ c\in G_{i+1},\exists\ s\subset c(i\text{-}subset),support(s)=support(c)\Rightarrow prune\ c\ from\ G_{i+1}$。一旦找出所有生成子，再扫描数据库一遍，确定它们的闭包，求出所有频繁闭合模式。

A-Close 基本过程与 Apriori 相同，即逐层生成、裁剪候选集，扫描数据库统计支持率。扫描数据库的次数一般比 Apriori 要少。由于搜索空间减小，效率大大提高。

4.2 对 A-Close 的改进

(1)Bastide 等改进 A-Close 算法，提出 Pascal 算法[11]，挖掘关键频繁模式，从中推出所有其它模式，效率比 A-Close 高2倍，比 Apriori 高10倍。

(2) Cristofor 等提出 Closure 算法[12]，效率略优于 Apriori。

4.3 闭合频集与无冗余关联规则研究的新进展

A-Close 及其改进算法，自底向上、宽度优先、逐层搜索、并剪裁搜索空间，效率有提高。但没有根本解决内在的模式匹配的高 CPU 开销和扫描数据库的高 I/O 开销。因此，很多研究提出了深度优先或混合搜索算法。

(1)Pei 等提出基于 FP-Growth 的 CLOSET 算法[28]，引入剪裁和检查“条件”数据库和 FP-Tree 的步骤，通过深度优先搜索来挖掘频繁闭合模式。其困难是递归构造“条件 FP-Tree”的 CPU 开销和存贮开销很大。另外，采用分割法处理大型数据集，检查局部频繁闭合模式的全局闭合性和频繁性的代价非常高。

(2)Zaki 等提出 CHARM 算法[33~35]，它采用垂直格式的事务标识集 tidset 来表示模式支持集，通过 IT 树实现模式集与事务标识集的双向搜索，搜索空间的剪裁效率很高。对于密集型数据集，采用 diffset 记录候选模式 tidset 与其父亲模式 tidset 的差异，减少内存开销。按局部支持率升序动态排列项目，提高剪裁机会。还采用基于 tidset 的杂凑函数来检查模式间的包含关系。其困难在于海量数据集无论采用 tidset 还是 diffset，存贮开销都非常大，投影效率不高，特别是密集型或存在特别长模式的数据集，其效率与可伸缩性仍受到了严重

挑战。

(3) Liu 等提出基于 OpportuneProject 的新算法 CROP[21]。它提出表达解空间的复合型频繁模式树,及与之相适应的解空间搜索效率与剪裁效率相平衡的原则。复合型频繁模式树与其它算法采用的字典树和概念格相比,结点数和层次少得多。所提出的局部和全局剪裁方法具有较高的解空间剪裁效率。其效率比 CHARM 高5倍以上,可伸缩性也高得多。

5 最大频繁模式挖掘

5.1 最大模式挖掘与 Pincer-Search 算法

最大模式是不被其它模式包含的频繁模式。Lin 等提出算法 Pincer-Search[19],以 Apriori 的自底向上搜索为基础,引入自顶向下搜索(超集频繁性剪裁)。

令 FREQ 和 INFREQ 分别是当前已知频繁模式和非频繁模式的总和。MFCS 是最大频繁模式候选集,概括 FREQ 的所有模式,排除 INFREQ 的所有模式。在自底向上搜索过程中,同时统计 MFCS 中模式的支持数,以尽早发现 MFCS 中所含的长频繁模式,实现自顶向下的超集频繁性剪裁。初始时,MFCS 只有一个元素,即所有项目的全集。第 k 次迭代时,新发现 MFCS 中长度大于 k 的频繁模式 X,则将 X 的所有 k 子集从 FREQ 中删除;新发现长度 k 的非频繁模式 Y,则将 MFCS 中所有包含 Y 的模式 X 用 $|Y|$ 个模式替换(通过从 X 中剔除一个 Y 中的项目)。MFCS 中模式互不包含,被包含者可从中删除。Pincer-Search 生成候选集时,不仅 FREQ 中保留的 k 频繁模式相互 join,它们还要与 MFCS 中各模式的 k 子集作 join。

Pincer-Search 减小了频繁模式的候选集,并且有可能减少数据库扫描。代价是存贮 MFCS,额外统计 MFCS 中模式的支持数,根据非频繁模式对 MFCS 进行更新。

5.2 其它基于 Apriori 的算法

(1) Bayardo 提出算法 MaxMiner[8],将最大频繁模式组织成集合枚举树。树结点 g 用候选组表示:head(g)是结点所表示的最大频繁模式,tail(g)包含可能出现在子女结点中的项目。扫描数据库时,不仅计算 head(g)的支持率,还统计 head(g)∪tail(g)、head(g)∪{i},∀ i∈tail(g)。如果 head(g)∪tail(g)频繁,则为局部最大频繁模式,分枝停止生长。如果 head(g)∪{i}非频繁,则将 i 从所有子结点中删去。如果 head(g)∪{i}频繁,则生成子女 g',head(g')=head(g)∪{i}。为提高 tail 项目被所有子结点 head 包含的可能性,充分利用频繁超集剪裁,对项目按局部支持率增序动态重新排列。此外,利用已知支持率信息,可计算子女支持率的下界,做进一步频繁超集剪裁。求出的局部最大频繁模式还要相互比较,确定全局最大频繁模式。

(2) Gunopulos 等提出一个随机算法[14],它试图不断扩大一个工作模式直到失败,每扩展一个模式就需要扫描数据库,因此需要在内存中存贮数据库,其伸缩性没有得到验证。它不能保证找出每个最大频繁模式。

(3) Zaki 等提出 MaxEclat 和 MaxClique 算法[37]。此两算法只在自底向上搜索的初始化阶段,试图提前发现最大频繁模式。MaxClique 采用基于动态规则的初始化,遇到较长的频繁模式时会有困难。

(4)最大模式挖掘算法还有 Gouda 等提出的 GenMax[15]等。

5.3 最大模式挖掘研究的新进展

近年来研究表明,要克服基于 Apriori 算法的缺陷,深度优先算法有很多优势。

(1) Agarwal 等提出深度优先挖掘最大频繁模式的 DepthProject 算法[2]。它采用选择性的投影,当投影数据库的尺寸低于某个阈值时,才在内存中实际建立结点的投影数据库,否则只保存一个二进制位串,指明先辈结点的投影数据库中支持当前结点的事务。在低层结点上,采用水平二进位串表示投影事务,存贮压缩率很高,而计数效率则是非压缩方法的8倍。然而,事务平均长度远小于不同项目的总数时,其效率比非压缩方法低。相反地,项目总数较少时,数据集的密度往往很大,其压缩率与基于树的表示法不相上下。而二进位串的预处理和后处理代价则是不可忽视的。其超集频繁性剪裁策略与 MaxMiner 类似,也需要最终剪裁步。

(2) Burdick 等提出算法 MAFIA[9],将频繁模式集组织成格。采用垂直二进制位图表示投影数据库。项目平均支持率高于1/32时,存贮效率高于非压缩方法,并且有选择地重建投影数据库。实现三种超集频繁性剪裁方法:tail 中支持率与 head 相同的项目直接转入 head;提前检查 head∪tail 的支持率;检查 head∪tail 是否已被其它模式包含。但与 DepthProject 类似,很难平衡投影重建与计数代价,并且预处理与后处理代价很高。据报道性能优于 DepthProject。

(3)刘等在 CROP 算法中引入一般性包含关系检查与剪裁,以及适时前窥措施,实现算法 MOP[39],其效率与可伸缩性优于 Maxminer 和 Mafia。

6 多维多层关联规则及其它

(1)多层关联规则挖掘问题。Agrawal 等提出的 Cumulate[30]和 Han 等提出的 ML-TmLn[16]最为著名。Cumulate 对每个事务按概念层次进行扩展,概念与项目一样处理,关键是保证概念层次中祖先后代不出现在同一个模式中,讨论了检测多层关联规则的冗余问题,可挖掘不同层之间的关联规则。ML-TmLn 采用自顶向下方式进行逐层挖掘,不支持不同层之间的关联规则挖掘,支持率阈值选择策略有二:固定阈值、自顶向下逐层递减的阈值。Cumulate 和 ML-TmLn 都是对单层 Apriori 的简单扩展,因此均存在与 Apriori 相同的缺陷。

(2)多维关联规则挖掘。挖掘对象是关系型数据库,其数据类型比较丰富,但可归结为数值型和类别型,数值型属性的区间分割是多维问题的关键。Srikant 等提出偏完全性概念[29],指出等深度区间分割信息损失最小。Lent 等提出用聚类法进一步聚合挖掘出的多维关联规则[22],以期得到更具普遍意义的规则。Miller 等提出基于距离的关联规则[24],主要思想是根据距离来分割数量型区间,以较好地反映属性的实际意义。Kamber 等提出直接利用数据仓库对属性进行静态分割和多维挖掘[18]。

(3)基于约束的关联规则挖掘。数据挖掘过程中可能发现许多与任务无关的规则,根据用户提供的各种类型的约束来指导挖掘,有利于提高挖掘的有效性与效率。约束类型主要有数据约束、维数与层次约束、规则形式约束等。Kamber 等讨论了采用元规则来表达约束[18]。Grahne 等将约束条件细分为单调、反单调、简约、可变换、不可变换的,分别讨论各类约束条件下剪裁搜索空间的策略[13]。

(4)并行挖掘。这是提高海量数据挖掘效率的途径之一。Zaki 等分别讨论在分布式存贮(零共享内存)并行机[36]和共享内存并行机[38]上,以 Apriori 框架,实现并行挖掘的方法。

(5)关联规则与相关性。对关联规则 $x \rightarrow y$,令 $lift(x \rightarrow y) = confidence(x \rightarrow y)/support(x)$,则 $lift$ 值有三种:等于1表明

x 与 y 无关，小于1表示 x 与 y 负相关，大于1表示 x 与 y 正相关。Brin、Ahmed 等均采用非支持率—可信度框架表达规则的意义[3,10]，提出了根据项目间统计相关性来挖掘相关性规则的算法。

(6)其它。关于多支持率规则挖掘[39]、增量型挖掘、周期性片断挖掘、周期性关联规则挖掘、事务数据库中的因果结构挖掘等问题亦有一些研究成果发表。

讨论 有关频集与关联规则挖掘的研究可从几个方面归纳。一是解空间类型，频集分为完全集、闭合集、最大集，关联规则分为单维单层规则、多维多层规则、多阈值规则、无冗余规则等；二是解空间表达形式，分为格、字典树、集合枚举树、频繁模式树等；三是解空间搜索策略，分为宽度优先、深度优先、混合搜索等；四是模式支持集表示形式，分为基于外存和基于内存两大类，后者包括基于树、基于数组、基于垂直二进位图、基于水平二进位串等；五是模式支持集确定及计数方法，基于外存的通过扫描数据库临时确定与计数，基于内存的通过投影来完成，其中基于树的可采用虚拟投影、基于数组的可采用非过滤投影、基于垂直二进位图的和基于水平二进位串的可采用选择性投影重建和桶计数法等。简言之，频集与关联规则挖掘的效率和有效性取决于解空间的表达形式、搜索策略，模式支持集的表达形式、确定(投影)和计数方法等因素。

有关研究已在多方面取得了重要进展，下一步研究重点是网络海量数据特别是流式数据的高性能挖掘算法和协同挖掘系统模型。

参考文献

1 Agarwal R, Aggarwal C, Prasad V V V. A tree projection algorithm for generation of frequent itemsets. In Journal of Parallel and Distributed Computing (Special Issue on High Performance Data Mining), 2000

2 Aggarwal C, Agarwal R, Prasad V V V. Depth First Generation of Long Patterns. In: The 6th ACM SIGKDD Intl. Conf. on Knowledge Discovery & Data Mining, Boston, MA, USA, 2000

3 Ahmed K M, El-Makky N M, Taha Y. A note on "Beyond market basket: Generalizing association rules to correlations." SIGKDD Explorations, 2000, 1: 46～48

4 Agrawal R, Imielinski T, Swami A. Mining association rules between sets of items in large databases. In SIGMOD'93, Washington, D. C., May 1993

5 Agrawal R, et al. Fast discovery of association rules. In: U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, R. Uthurusamy, eds. Advances in Knowledge Discovery and Data Mining, AAAI/MIT Press, 1996. 307～328

6 Agrawal R, Srikant R. Fast algorithms for mining association rules: [Reearch Report RJ 9839]. IBM Almaden Research Center, San Jose, CA, June 1994

7 Agrawal R, Srikant R. Fast algorithms for mining association rules. In VLDB'94, Santiago, Chile, Sept. 1994. 487～499

8 Bayardo R J. Efficiently mining long patterns from databases. In SIGMOD'98, Seattle, Washington, June 1998. 85～93

9 Burdick D, Calimlim M, Gehrke J. MAFIA: a maximal frequent itemset algorithm for transactional databases. In: Intl. Conf. on Data Engineering, Apr. 2001

10 Brin S, et al. Dynamic Itemset Counting and Implication Rules for Market Basket Analysis. In SIGMOD'97, Tucson, AZ, May 1997. 255～264

11 Bastide Y, Taouil R, Pasquier N, et al. Mining frequent patterns with counting inference. SIGKDD Explorations, 2000, 2(2)

12 Cristofor D, Cristofor L, Simovici D. Galois connection and data mining. Journal of Universal Computer Science, 2000, 6(1): 60～73

13 Grahne G, Lakshmanan L, Wang X. Efficient mining of constrained correlated sets. In: Proc. 2000 Int. Conf. Data Engineering (ICDE'00), San Diego, CA, Feb. 2000. 512～521

14 Gunopulos D, Mannila H, Saluja S. Discovering all Most Specific Sentences by Randomized Algorithms. In: 5th Int'l Conf. on Database Theory, Jan. 1997

15 Gouda K, Zaki M J. Efficiently mining maximal frequent itemsets. In: 1st IEEE Int'l Conf. on Data Mining, Nov. 2001

16 Han J, Fu Y. Discovery of multiple-level association rules from large databases. In VLDB'95, Zuich, Switzerland, Sept. 1995. 420～431

17 Han J, Pei J, Yin Y. Mining frequent patterns without candidate generation. In SIGMOD'2000, Dallas, TX, May 2000

18 Kamber M, Han J, Chiang J Y. Meta-rule-guided mining of multi-dimensional association rules using data cubes. In: Proc. 1997 Int. Conf. Knowledge Discovery and Data Mining (KDD'97), Newport Beach, CA, Aug. 1997. 207～210

19 Lin D-I, Kedem Z M. Pincer-search: A new algorithm for discovering the maximum frequent set. In: 6th Intl. Conf. Extending Database Technology, March 1998

20 Liu Junqiang, Pan Yunhe, Wang Ke, Han Jiawei. Mining frequent item sets by opportunistic projection. In: Proc. of the Eighth ACM SIGKDD Intl. Conf. on Knowledge Discovery and Data Mining, Alberta, Canada, July, 2002. 229～238

21 Liu Junqiang, Pan Yunhe. An efficient algorithm for mining closed itemsets. Journal of Zhejiang University SCIENCE (Accepted in Feb. 2003)

22 Lent B, Swami A, Widom J. Clustering association rules. In: Proc. 1997 Int. Conf. Data Engineering (ICDE'97), Birmingham, England, Apr. 1997. 220～231

23 Mannila H, Toivonen H, Verkamo A I. Efficient algorithms for discovering association rules. In: Proc. AAAI'94 Workshop Knowledge Discovery in Databases (KDD'94), Seattle, WA, July 1994. 181～192

24 Miller R J, Yang Y. Association Rules over Interval Data. In SIGMOD'97, Arizona, USA, 1997. 425～461

25 Pasquier N, Bastide Y, Taouil R, Lakhal L. Pruning closed itemset lattices for association rules. In: Proc. of the BDA French Conf. on Advanced Databases, Oct. 1998

26 Pasquier N, Bastide Y, Taouil R, Lakhal L. Discovering frequent closed itemsets for association rules. In ICDT'99, Jerusalem, Israel, Jan. 1999. 398～416

27 Park J S, Chen M S, Yu P S. An effective hash based algorithm for mining association rules. In: Proc. 1995 ACM-SIGMOD, San Jose, CA, Feb. 1995. 175～186

28 Pei J, Han J, Mao R. CLOSET: An Efficient Algorithm for Mining Frequent Closed Itemsets. In: ACM-SIGMOD Workshop on Data Mining and Knowledge Discovery (DMKD'00), Dallas, TX, May 2000

29 Srikant R, Agrawal R. Mining quantitative association rules in large relational tables. In SIGMOD'96, Montreal, Canada, June 1996. 1～12

30 Srikant R, Agrawal R. Mining generalized association rules. In VLDB'95, Zurich, Switzerland, Sept. 1995. 407～419

31 Sarasere A, Omiecinsky E, Navathe S. An efficient algorithm for mining association rules in large databases. In: 21st Int'l Conf. on Very Large Databases (VLDB), Zurich, Switzerland, Sep. 1995

32 Toivonen H. Sampling large databases for association rules. In: Proc. 1996 Int. Conf. Very Large Data Bases (VLDB'96), Bombay, India, Sept. 1996. 134～145

33 Zaki M J. Generating Non-Redundant Association Rules. The 6th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Boston, MA, Aug. 2000. 34～43

34 Zaki M J, Hsiao C-J. CHARM: An efficient algorithm for closed association rule mining: [Technical Report 99-10]. Computer Science Dept., Rensselaer Polytechnic Institute, Oct. 1999. 28

35 Zaki M J, Hsiao C-J. CHARM: An Efficient Algorithm for Closed Itemset Mining. The 2nd SIAM Intl. Conf. on Data Mining, Arlington, VA, USA, April 2002

36 Zaki M J, Parthasarathy S, Ogihara M, Li W. Parallel algorithms for discovery of association rules. Data Mining and Knowledge Discovery, 1997, 1(4): 343～374

37 Zaki M J, Parthasarathy S, Ogihara M, Li W. New Algorithm for Fast Discovery of Association Rules. In: Proc. of the 3rd Int'l Conf. on Knowledge Discovery in Databases and Data Mining, 283～286

38 Zaki M J, Parthasarathy S, Ogihara M, Li W. Parallel data mining for association rules on shared-memory systems. Data Mining and Knowledge Discovery, 0: 1-32, 1998

39 刘君强. 海量数据挖掘研究: [浙江大学博士学位论文]. 杭州: 浙江大学计算机科学与技术学院, 2003